

Amendments to the Claims:

This listing of the claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (previously presented) A method for generating a reflection of data in a plurality of processing elements, comprising:
 - shifting the data along either the rows or columns of the plurality of processing elements arranged in an $N \times N$ array, where N is greater than three until each processing element in each row or column has received the data originally held by every other processing element in that row or column, respectively; and
 - selecting from said received data, where each of the received data is a candidate for selection, a final output based on a processing element's position.
2. (original) The method of claim 1 additionally comprising one of loading an initial count into each processing element and calculating an initial count locally based on the processing element's location, said selecting being responsive to said initial count.
3. (previously presented) The method of claim 2 wherein said plurality of processing elements is arranged in an array and said initial count is given by the expression $(2 \times \text{Col_Index}) \text{ MOD } (\text{array size})$, where array size equals N .
4. (previously presented) The method of claim 2 wherein said plurality of processing elements is arranged in an array and said initial count is given by the expression $(2 \times \text{Row_Index}) \text{ MOD } (\text{array size})$, where array size equals N .
5. (original) The method of claim 2 additionally comprising maintaining a current count in each processing element, said current count being responsive to said initial count and the number of data shifts performed, said selecting being responsive to said current count.
6. (original) The method of claim 5 wherein said maintaining a current count includes altering said initial count at programmable intervals by a programmable amount.
7. (original) The method of claim 5 wherein said initial count is decremented in response to said shifting of data to produce said current count.
8. (original) The method of claim 6 wherein said selecting occurs when said current count is non-positive.

9. (original) The method of claim 1 additionally comprising maintaining a local count including setting a counter to a first known value, and counting up from said first known value based on the number of shifts that have been performed, said selecting occurring when a current count equals a target count.

10. (original) The method of claim 1 wherein said shifting includes a wrap shift.

11. (previously presented) The method of claim 10 wherein said wrap shift includes shifting data west to east and east to west along rows, or includes shifting data north to south and south to north along columns.

12. (previously presented) A method for generating a reflection of data in an array of processing elements, comprising:

shifting originally held data along either the rows or columns in the array a number of times equal to $N-1$ where N equals the number of processing elements in the rows and columns, respectively, and N is greater than three, such that each processing element in each row or column receives the data originally held by every other processing element in that row or column, respectively; and

outputting data from each processing element as a function of that element's position in one of the row and column, where each of the received data is a candidate for outputting.

13. (original) The method of claim 12 additionally comprising one of loading an initial count into each processing element and calculating an initial count locally based on the processing element's position in one of the row and column, said outputting being responsive to said initial count.

14. (previously presented) The method of claim 13 wherein said initial count is given by the expression $(2 \times \text{Col_Index}) \bmod (\text{array size})$, where array size equals N .

15. (previously presented) The method of claim 13 wherein said initial count is given by the expression $(2 \times \text{Row_Index}) \bmod (\text{array size})$, where array size equals N .

16. (previously presented) The method of claim 13 additionally comprising maintaining a current count in each processing element, said current count being responsive to said initial count and the number of data shifts performed, said outputting being responsive to said current count.

17. (original) The method of claim 16 wherein said maintaining a current count includes altering said initial count at programmable intervals by a programmable amount.

18. (original) The method of claim 16 wherein said initial count is decremented in response to said shifting of data to produce said current count.
19. (original) The method of claim 18 wherein said outputting occurs when said current count is non-positive.
20. (original) The method of claim 12 additionally comprising maintaining a local count including setting a counter to a first known value, and counting up from said first known value based on the number of shifts that have been performed, said outputting occurring when a current count equals a target count.
21. (original) The method of claim 12 wherein said shifting includes a wrap shift.
22. (previously presented) The method of claim 21 wherein said wrap shift includes shifting data west to east and east to west along rows, or includes shifting data north to south and south to north along columns.
23. (currently amended) ~~A method for reflecting data in a matrix of processing elements about a line, comprising:~~ The method of claim 2 wherein said initial count is set
~~— shifting data between processing elements arranged in an array of rows and columns;~~
~~— setting an initial count in each processing element according to one of the expressions~~
$$(2 \times \text{Col_Index}) \bmod (\text{array size})$$

$$(2 \times \text{Row_Index}) \bmod (\text{array size});$$

where array size equals N for an $n \times n$ array and n is greater than 3; said method additionally comprising
modifying said initial count by a programmable amount at programmable intervals to produce a current count; and wherein
said selecting is responsive to output data as a function of said current count.
24. (original) The method of claim 23 wherein said modifying includes counting down from said current count.
25. (original) The method of claim 24 wherein said selecting occurs when said current count is a non-positive value.
26. (original) The method of claim 23 wherein said shifting includes a wrap shift.
27. (previously presented) The method of claim 26 wherein said wrap shift includes shifting data one of east to west, west to east, north to south and south to north.

28. (previously presented) A computer readable storage medium carrying a set of instructions which, when executed, perform a method comprising:

shifting data along either rows or columns of a plurality of processing elements arranged in an $N \times N$ array, where N is greater than three until each processing element in each row or column has received all the data originally held by every other processing element in that row or column, respectively; and

selecting from said received data, where each of the received data is a candidate for selection, a final output based on a processing element's position.

29. (new) A computer readable storage medium carrying a set of instructions which, when executed, perform a method comprising:

shifting originally held data along either rows or columns in an array a number of times equal to $N-1$ where N equals the number of processing elements in the rows and columns, respectively, and N is greater than three, such that each processing element in each row or column receives the data originally held by every other processing element in that row or column, respectively; and

outputting data from each processing element as a function of that element's position in one of the row and column, where each of the received data is a candidate for outputting.